Modeling Unmanned Vehicle System

Ser Keong Lim, Chua Ching Hao
Purdue University, Department of Aeronautics and Astronautics

Abstract

The main objective of this research is to obtain a realistic model of an Unmanned Vehicle System. The method and procedures used are compiled below. The model of the aircraft will be made based on the given data/information and finally the aircraft will be analyzed. The main output data from the research will be the trim conditions of the aircraft in different flight condition.

What had been done in Spring 2012?

1) Finalizing the JSBSim graphical interphase window
2) Revising the Shadow Model
3) Created the numeric_analysis of Moments of Inertia (MOI) of an aircraft
4) Obtain the output of JSBSim for Shadow UR-7
5) Modeling of the missile using missile DATCOM+ (adaptation taken from X-15)
6) Modeling the entire orbiter
7) Modeling the entire Aerosonde
8) Start on Cargo-UAS (incomplete, need to be continued next semester)

Moments of Inertia of an Aircraft

The moments of inertia of an aircraft are crucial information which affects the rotation of the aircraft in the 3 axis. Hence, basic estimated relationships between the weight of the aircraft to the value of moments of inertia Ixx, Iyy and Izz are made. Real data on the weight and moments of inertia of a few aircraft are obtained. The plots of moments of inertia are made with respect to the weight of the aircraft. Polynomials of 3 are used to estimate the relationship.

The equations are: (y is moment of inertia(slug ft^2) ; x is weight (lbs))

$$Ixx: 1.50936e\text{-}7x^3 - 0.000145021x^2 + 0.535851x \quad -(1)$$
$$Iyy: 1.24339e\text{-}7x^3 - 0.000120154x^2 + 0.655246x \quad -(2)$$
$$Izz: 4.31686e\text{-}7x^3 - 0.0007283x^2 + 1.32361x \quad -(3)$$

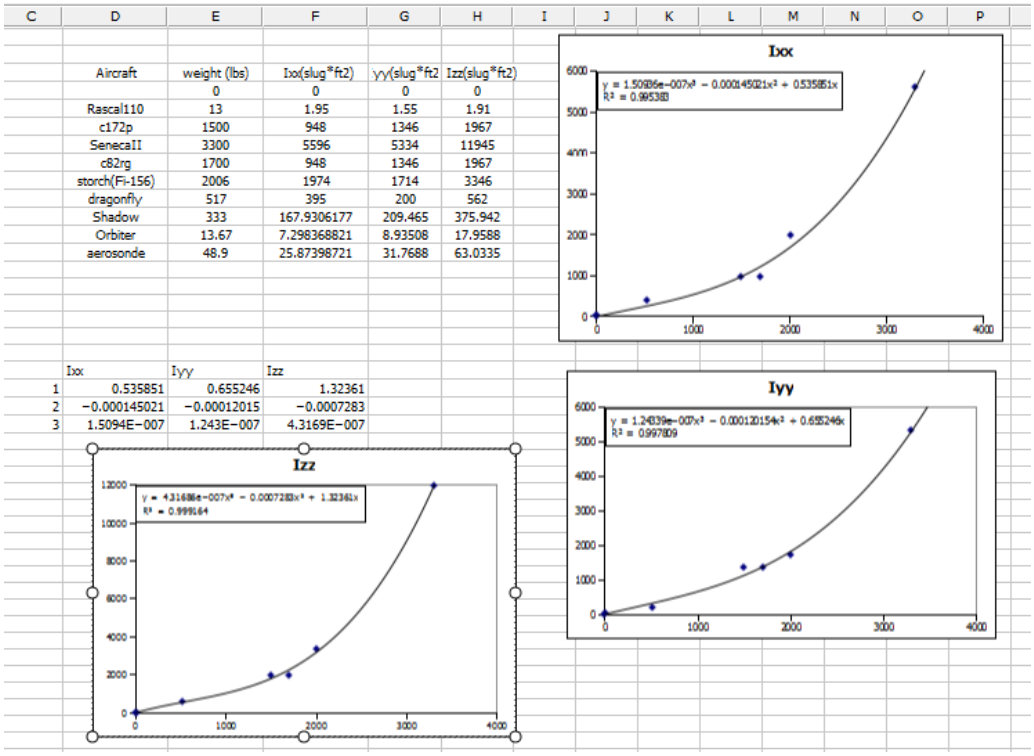

Figure 1: Moment of Inertia Computation table/chart

The picture depicts 3 plots obtained in Microsoft excel. These estimations are quite close to the real values.

Modeling of Unmanned Aircraft

In the entire process of modeling the unmanned aircraft, there are 4 main programs that will be used throughout; there are Blender, DATCOM+, FLightGear, and JSBSim. This report will entirely explain in detail the entire process.

File Storing and location

Basically the file for the folder will be the name of the aircraft itself.
The figure below is the explorer on the content of an example aircraft (shadow):

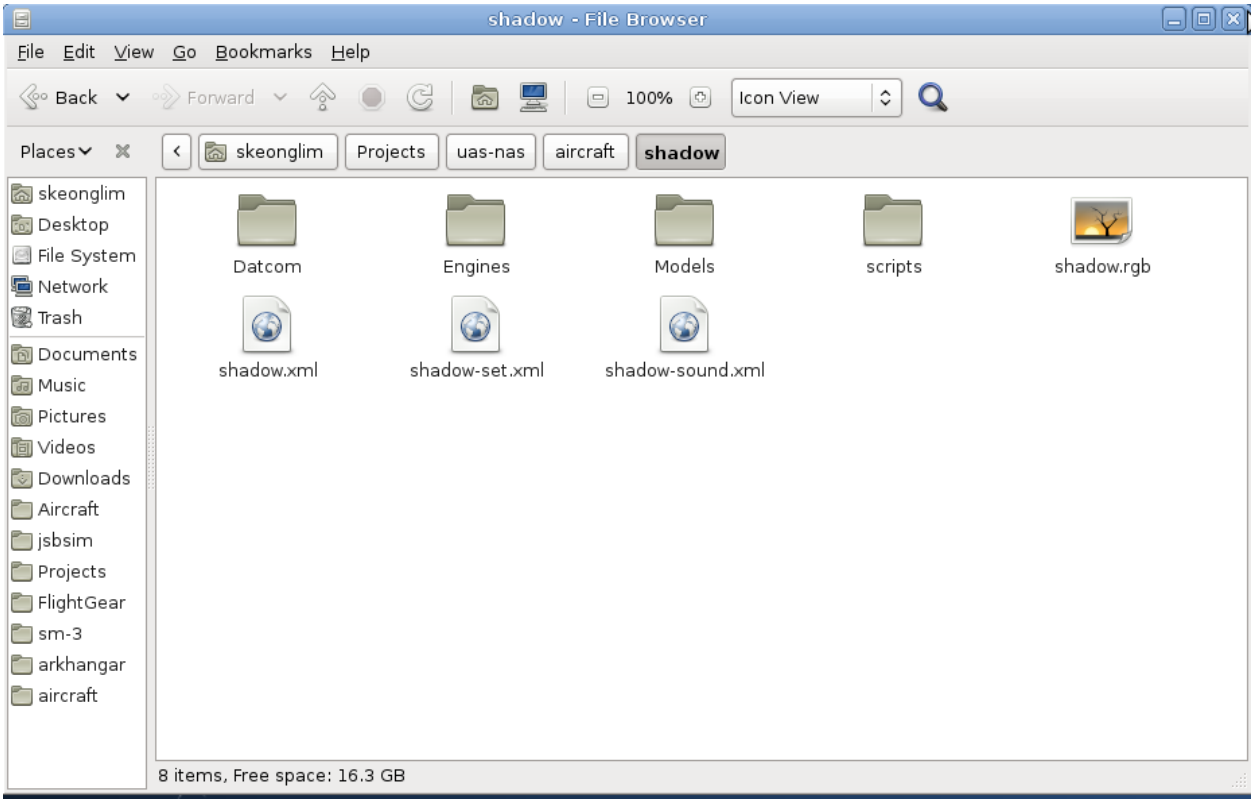

Figure2: File location (Example used: shadow)

The basic explanations of the purposes and contents of each file:
(Note: “aircraft_name” stands for the aircraft’s name, in this case “shadow”)

1) DATCOM+ (file)
    a. Contains all files that will be used by DATCOM+ to generate the aerodynamics output of the aircraft model.
    b. The main output of DATCOM+ will be the aircraft_name_aero.xml, which will be accessed later in aircraft_name.xml file.
2) Engines (file)
    a. Contains the engine file (xml type) that will be accessed later in aircraft_name.xml file.
3) Models (file)
    a. Contains all files that will be used by Blender which produces the 3D model of the aircraft.
    b. Basically contains 4 files which are:
        i. aircraft_name-m.blend – aircraft in meters.
        ii. aircraft_name.bend – aircraft in feet.
        iii. aircraft_name.ac – AC3D model of aircraft.

iv. aircraft_name_model.xml contains the animations path and the AC3D model that will be accessed in aircraft_name.xml file

4) Scripts (file)
    a. Contains all different flight conditions of aircraft.
5) Shadow.rgb (rgb)
    a. A picture of shadow that shown in FlightGear.
6) Shadow (xml)
    a. Main file which contains all the data of the aircraft (detail explanation will be given below).
7) Shadow-set (xml)
    a. First file that are accessed by FlightGear and which direct all the necessary paths including the main file, shadow.xml file.
8) Shadow-sound (xml)
    a. The aircraft sound file.

The examples on how the following file and uses will be explained below.

Main Procedure for Modeling (brief, the order is advisable)

1) Obtain all required aircraft information
2) Copy entire file from the similar model:
    a) Previous model (Arkhangar, Easystar, Shadow, Aerosonde, Orbiter)
    b) Or reference from model from FlightGear
3) Edit the name and change the paths for all in the aircraft_name-set.xml
4) Edit name and change all the paths for all in the aircraft_name.xml
5) Use Blender to obtain the model
6) Use Datcom to obtain the aerodynamics data
7) Use FlightGear to test the aircraft model
8) Use JSBSim to obtain all required output
9) Edit Spring Coefficient
10) Edit aero.xml

Blender

Blender is a free open source 3D modeling site that can be downloaded at http://www.blender.org/download/get-blender/. Blender is able to run at most of the common used operating systems. Blender is used in the project this semester to replicate the model of the aircrafts by the provided dimensions from the given spreadsheets. A useful shortcut key is provided below, and the method used to replicate the desired models in Blender in this semester is discussed as followed.

Importance of shortcut keys

In Blender, unlike many other 3D modeling tools, shortcut keys are vital and used frequently to achieve best proficiency. A list of useful shortcut keys is provided in Table 1 as future references. A full list of shortcut keys in Blender can also be found at http://www.katsbits.com/tutorials/blender/useful-keyboard-shortcuts.php

Table 1: Useful Shortcut Keys in Blender

Shortcut Key	Description
/	Specifying on the interest part
A	Selecting all
M	Choosing of layer
N	Transformation properties
spacebar	Properties
S	Scale
s→x	Scale in x-direction
s→y	Scale in y-direction
s→z	Scale in z-direction
P	Make Parent
1-9(number pad)	View at different directions
Ctrl+z	Undo
Ctrl+d	Duplicate (all functions)
Shift+d	Duplicate
`	Show all components in all layers
Tab	Switch between Edit and Object Mode

An example of the function ‘ ` ’ in the list of shortcut key is shown below (see figure 3),

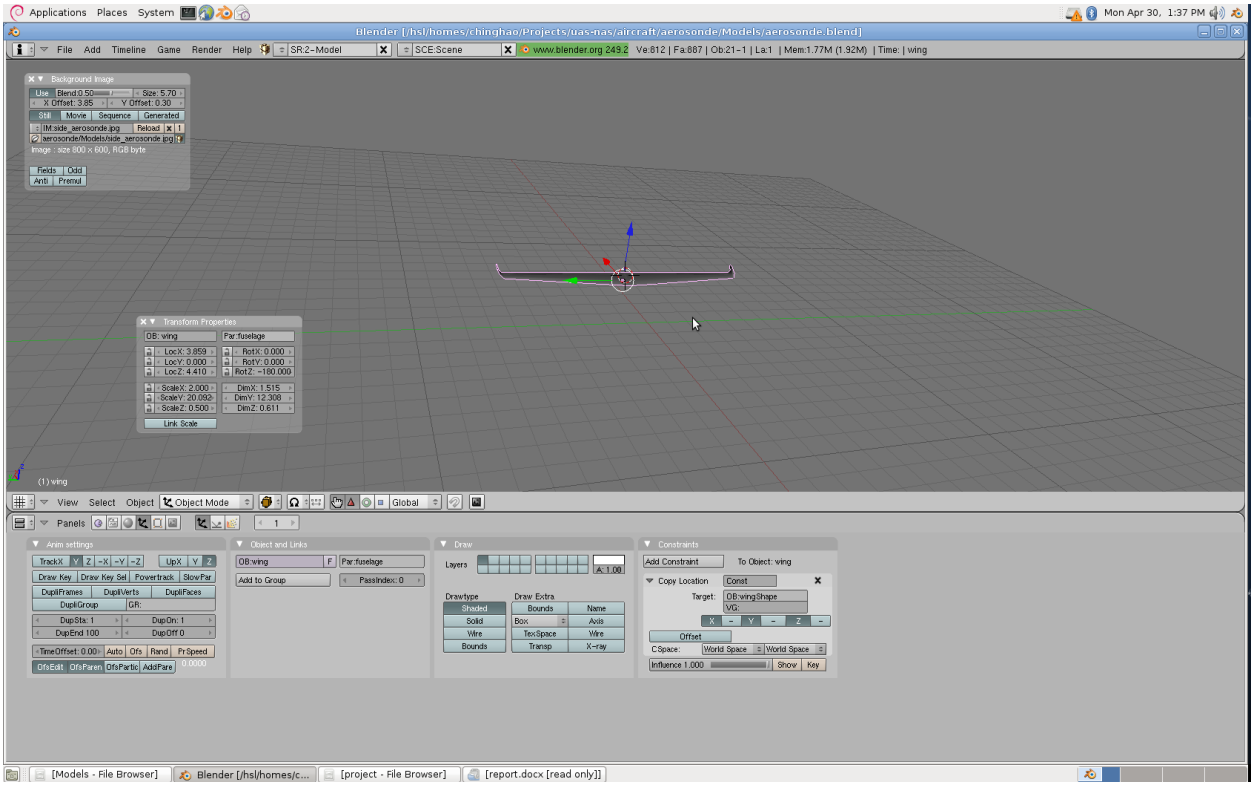

Figure 3: The demonstration of ‘ ` ’ function.

Drawing Procedures used in this semester.

Drawing procedures used in this semester are discussed in this section. For the ease of explanation, aircraft Aerosonde will be used as an example to better explain the procedures of modeling Aerosonde. The procedures discussed in below can be applied to future aircrafts that shall be modeled.

1. Search the corresponding side-view and top-view images of Aerosonde.
   Import the found Aerosonde images to Blender as the background images at the correct views.
2. Sketch of Aerosonde is started at the origin. Mirror images are always used for the ease of constructing the model.
3. Scale Aerosonde to the size of the background images.

Note: The inserted background images are for the use of reference. The interested model should always be drawn according to the dimensions provided in the given spreadsheet.

Input of Background Image

One of the basic steps before started the modeling of the desired aircraft is to input a precise top-view and side-view image of the actual model. This is done for a rough estimation of the scale of the model. An example is shown as below (see figure 4).

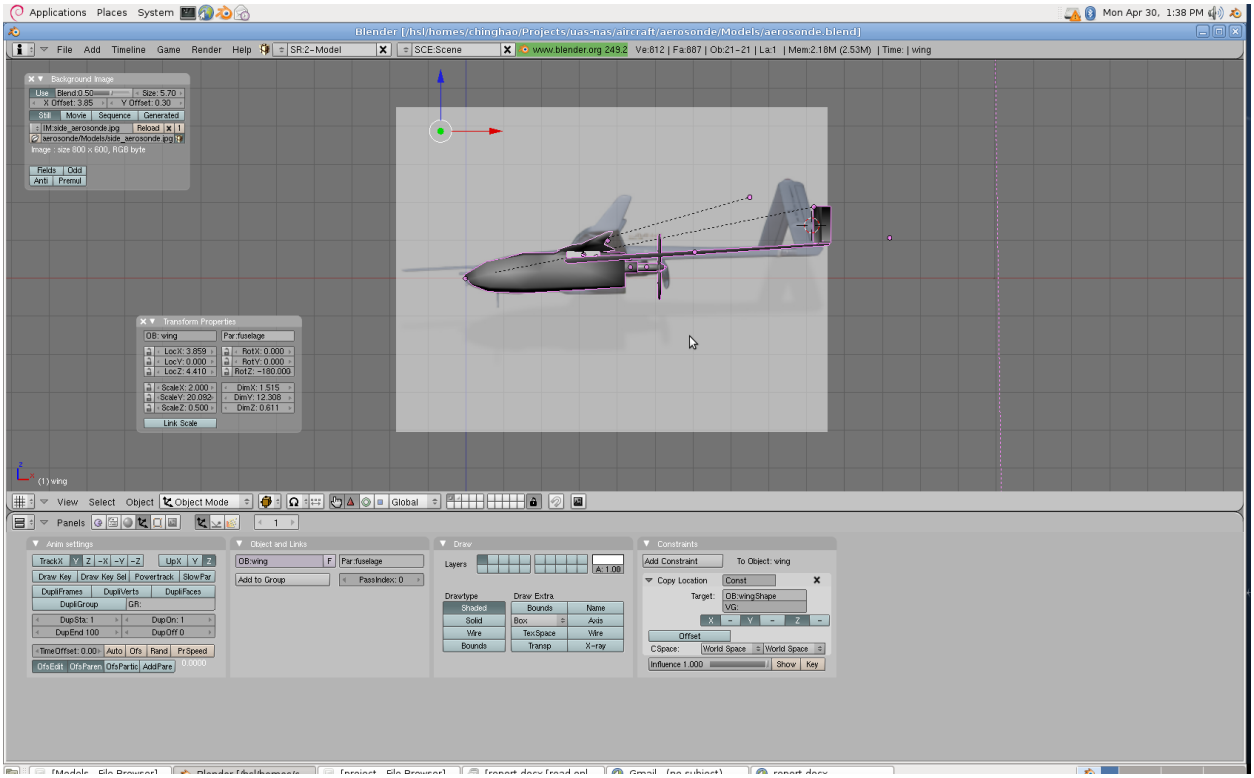

Figure 4: The insert of the background image (sideview)

As can be observed from figure 4, images at different view-angles are inserted for the users' ease to model the aircrafts as accurate as possible to the given dimensions. Users are able to compare their sketching models to the actual images inserted to prevent significant errors in the modeling from happening.

Modifications of Aerosonde

Aerosonde is built on top of Shadow. The process of modification from Shadow to Aerosonde will be discussed below. The basic steps are highlighted and more explanations can be found on the "details" section.

Steps of constructing winglets

Winglets are added by playing around with the extrude function. The following steps explain the method of extruding a desirable plane.

1. On the "Object Mode", select the desirable component.
2. Change the "Object Mode" to be "Edit Mode" by pressed "tab" once.

3. Select “Face Select” before select the desirable plane.
4. Select the desirable plane.
5. Extrude the plane by shifting 3D axis shown on the screen in the desirable direction.

Steps of constructing Flaps and Ailerons

The adjustments of flaps and ailerons can be done easily by resizing the dimensions of the flaps and ailerons. However, the resizing process can be hazardous if it is done improperly*. Thus, the constructions of flaps and ailerons will be introduced below for the future references. The following highlighted steps explain the ideas of constructing flaps and ailerons. More explanations on each step can be found in the remarks if necessary. The ways of constructing ailerons are not discussed below because they follow the same logic as the construction of flaps.

1. The construction of the cube on the second layer.
2. The move of the cube to the first layer.
3. Renamed of the cube and the addition of modifier to the cube.
4. Addition of modifiers to the “wing”.
5. The construction of the “leftFlap”.
6. Addition of constraint to the “leftFlap”.
7. Completion of the build of “leftFlap”.

Details:

1. On the second layer, a cube of the same dimension to the length of the flaps provided in the spreadsheet is constructed, it is then moved to the correct locations on the positive y-axis. The cube is named to be “flapCut”
2. Duplicate the constructed cube on the second layer and move it to the first layer, the above instructions can be done by following “Ctrl+d” → “m” → “1” → “Enter”.
3. On the first layer, “flapCut” is renamed to be “rightFlap”. Modifier “Boolean” shall be applied to “rightFlap”. The object is chosen to be “wingShape” and the option is chose to be “Difference” to substract “wingShape” from the modified mesh.
4. On the body “wing”, modifier “Boolean” shall also be applied. The object is chosen to be “rightFlap” and the option is chosen to be “Intersect”.
5. To replicate flap to the left wing, a duplication of the “rightFlap” shall be first done and be renamed to “leftFlap”.
6. A constraint “copylocation” is added to “leftFlap”, target is chosen to be “rightFlap” to duplicate the location of the “rightFlap”.
7. “-“on the right of “y” is pressed to replicate it on the negative y-axis.

*Note: Scaling in the size of any object without shifting the center of the object remains stationary is risky. The center of the object remains at the same position although the object becomes asymmetric because of the rescaling process.

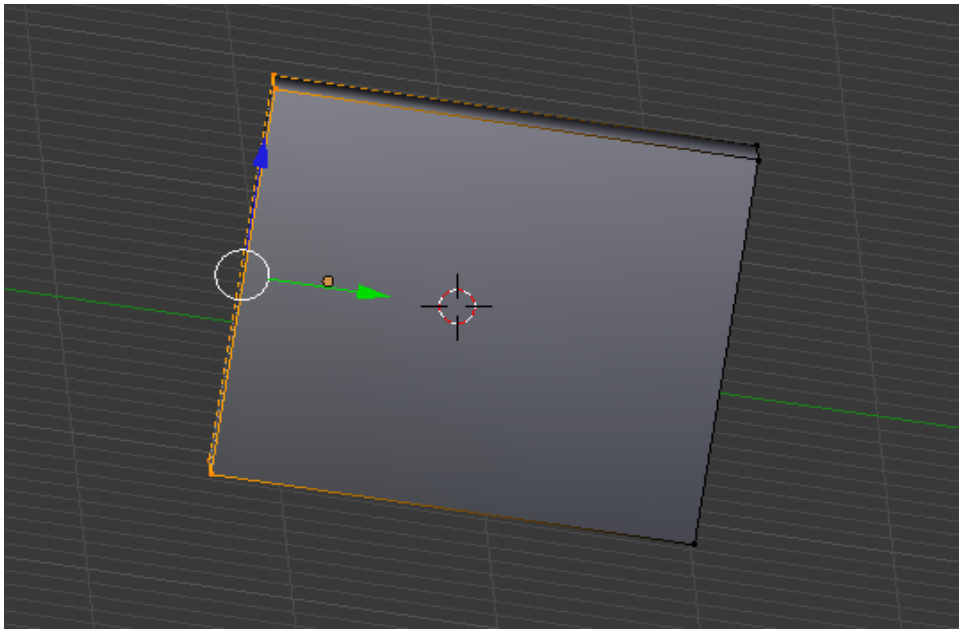

Figure 5: Object with center offset.

As can be seen from figure 5, the original center is the orange color dot. However, after the rescaling of one side of the object, the center is required to be reset to have the object to be symmetric about the center of the object.

Elevators and Rudder

Elevators and rudder are constructed in very similar ways to the constructions of flaps and ailerons. However, modifier “Union” is chosen instead of “Difference” which used in the constructions of flaps and ailerons on the wing. The following highlighted steps are advisable to be followed for the constructions of the elevators. More explanations can be found in “Details” if necessary.

1) The construction of the cube on the second layer.
2) The move of the cube to the first layer.
3) Renamed of the cube and the addition of modifier to the cube.
4) Addition of modifiers to the “tail”.
5) The construction of the “rightRuddervator”.

6) Addition of constraint to the “leftRuddervator”.
7) Completion of the build of “leftRuddervator”.

Details:

1) On the second layer, a cube of the same dimension to the length of the flaps provided in the spreadsheet is constructed, it is then moved to the correct locations on the positive y-axis. The cube is named to be “ruddervatorCut”
2) Duplicate the constructed cube on the second layer and move it to the first layer, the above instructions can be done by following “Ctrl+d” → “m” → “1” → “Enter”.
3) On the first layer, “ruddervatorCut” is renamed to be “rightRuddervator”. Modifier “Boolean” shall be applied to “rightRuddervator”. The object is chosen to be “wingShape” and the option is chosen to be “Difference” to subtract “tailShape” from the modified mesh.
4) On the body “tail”, modifier “Boolean” shall also be applied. The object is chosen to be “rightRuddervator” and the option is chosen to be “Union” to combine two meshes in an additive way.
5) To replicate rudder to the left wing, a duplication of the “rightRuddervator” shall be first done and be renamed to “leftRuddervator”.
6) A constraint “copylocation” is added to “leftRuddervator”, target is chosen to be “rightRuddervator” to duplicate the location of the “rightRuddervator”.
7) “-“on the right of “y” is pressed to replicate it on the negative y-axis.

Importance of having consistency in the naming process.

For every component created in Blender, an arbitrary name is generated automatically associated with the created component. At the ease of future modifications on the model and the creation of the animation of the model in FlighGear, the names of each component are crucial and shall be in the same consistency (see figure 6). The idea of having the first letter of the second word to be capitalized was implemented in this semester to prevent confusions in the names of each component. An example of the naming procedure is shown as below.

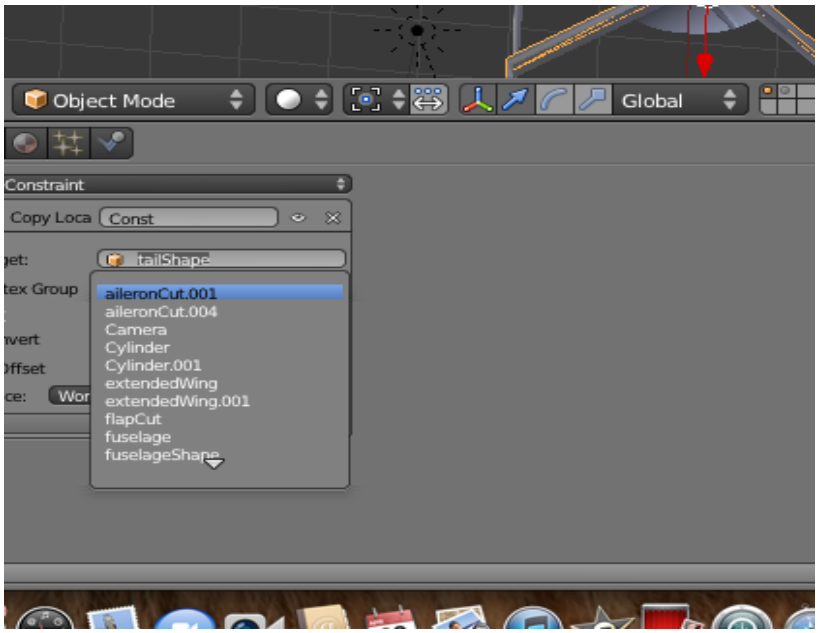

Figure 6: Sample Names of Components.

From the above figure, flap at the right side of the aircraft (on the positive y-direction) is named as rightFlap. The aileron right beside the rightFlap is named as rightAileron. Such analogy in the naming of the Aerosonde's components is also implemented for the other components of the Aerosonde.

Export method

Ac3d is used and supported by FlightGear simulator. In order to export the completed part into the desired modeler Ac3d, the methods that will be discussed below shall be followed. By different provided spreadsheet information, some of the aircrafts are sketched in feet or meter or other different units. FlighGear adapts "meter" in recreating the sketched model. Hence, the unit of the output Ac3d (.ac) file shall be changed to meter if it was not. If the model wasn't sketched in "meter", the following procedures shall be followed in order to change the unit of the model to be in "meter".

1. Selecting all components of aircraft.
2. Exports the file to Ac3d (.ac).
3. Open a new .blend file.
4. Import the previously saved .ac file
5. Scale the aircraft to desirable unit.
6. Saves and exports the file to Ac3d (.ac) again.
7. Remove the unnecessary aircraft-1.blend files.

Details:

1. At the object mode, select the entire model by pressing “a” once (see figure 7).
2. File →Export → .ac, saves the Ac3d (.ac) file in the correct folder with logical name.
3. File → New → “Delete the existing” → deletes the cube that is shown in the new page of Blender.
4. File → Import → .ac, find the saved Ac3d (.ac) file and import it to the present page by clicking the desired file.
5. Scaling the entire model to the “meter” by multiplying the model with the correct unit conversion. For example, to scale a model from “Feet” to “Meter”, one has to be multiplying by 0.3048 to have the right transformation.
6. Saves the newly transformed file to aircraft-m.blend and exports it to Ac3d (.ac) file again.
7. Inside of the model folder, files that are named “aerosonde-1.blend” are the associated back-up files that generated by Blender automatically. It shall be removed to avoid confusion.

A figure of all components of Aerosonde being selected in Object Mode is shown below,

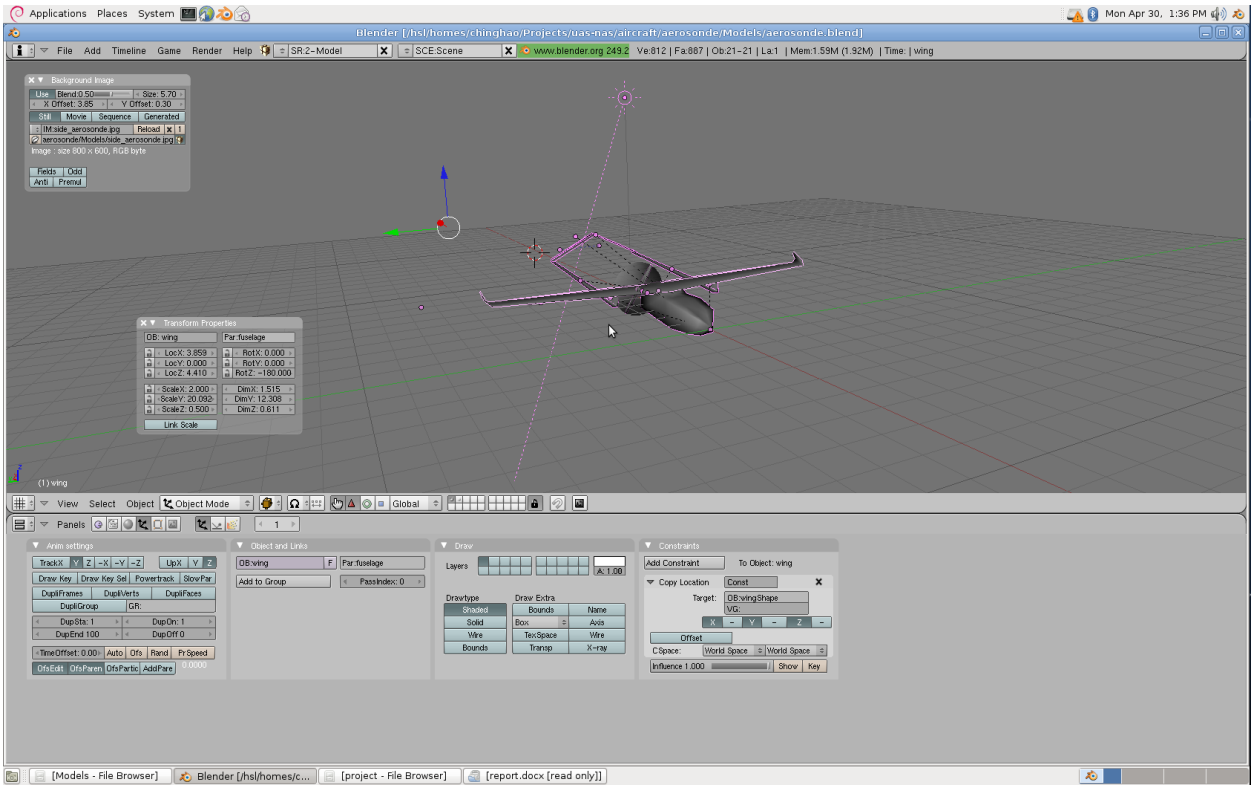
Applications Places System
Mon Apr 30, 1:36 PM
File Add Timeline Game Render Help
View Select Object Object Mode
[Models - File Browser]
[project - File Browser]
[report.docx [read only]]

Figure 7: Aircraft model with all components being selected (Aerosonde)

<u>Change of Names of Components in aircraft-model.xml</u>

After the completion of the modeling of aircraft in Blender, users shall check the names of each component in Blender match the names in “aircraft-model.xml.” Names in “aircraft-model.xml” are crucial for the animations of the parts of aircraft. Furthermore, users are also required to set the values associated to each of the name in “aircraft-model.xml” according to the Transform Properties given in Blender because these values affect the outcomes of the animations significantly.

```
<!--Ruddervators-->
<animation>
    <type>rotate</type>
    <object-name>leftRuddervator</object-name>
    <property>/surface-positions/left-ruddervator-pos-deg</property>
<axis>
        <x1-m>2.215</x1-m>
        <y1-m>-0.607</y1-m>
        <z1-m>0.178</z1-m>
    <x2-m>2.215</x2-m>
        <y2-m>-0.02</y2-m>
        <z2-m>0.796</z2-m>
    </axis>
</animation>
```

Figure 8: Name and Values Associated in aerosonde-model.xml

From figure 5, it can be observed that the “object-name” is named exactly the same as the way it is named in Blender. The values that are named “x1-m”, “y1-m”, and “z1-m” are the values of the furthest left point on the left rudder. While the values that are named “x2-m”, “y2-m”, and “z2-m” are the values of the furthest right point on the left rudder.

DATCOM+

Introduction

DATCOM is a free program that produces the aerodynamics data of the aircraft model. It is very user friendly.

The outputs of the following program are mainly:

a) Various aerodynamic graphs (jiff format)
b) Output aerodynamics data, aircraft_name.out (which consist of the numerical data calculated based on the parameters of the aircraft input).
c) Aerodynamics data, aircraft_name_aero.xml which will be used for aircraft_name.xml

As it is a free program accessible to everyone, the link to download the program is http://www.holycows.net/Datcom/. The user manual document is available in the following website for future references.

Procedure

1) Open aircraft_name.dcm file
2) Input the dimensions of the aircraft based on the model that are made in Blender
3) Basic body properties include:
    - Wing,
    - Flaps
    - Aileron
    - Elevator
    - Flaps
    - Horizontal tail
    - Vertical tail
4) To make file, type command “make” in the terminal. (Makefile file defines the command)

Remarks:

vi Makefile

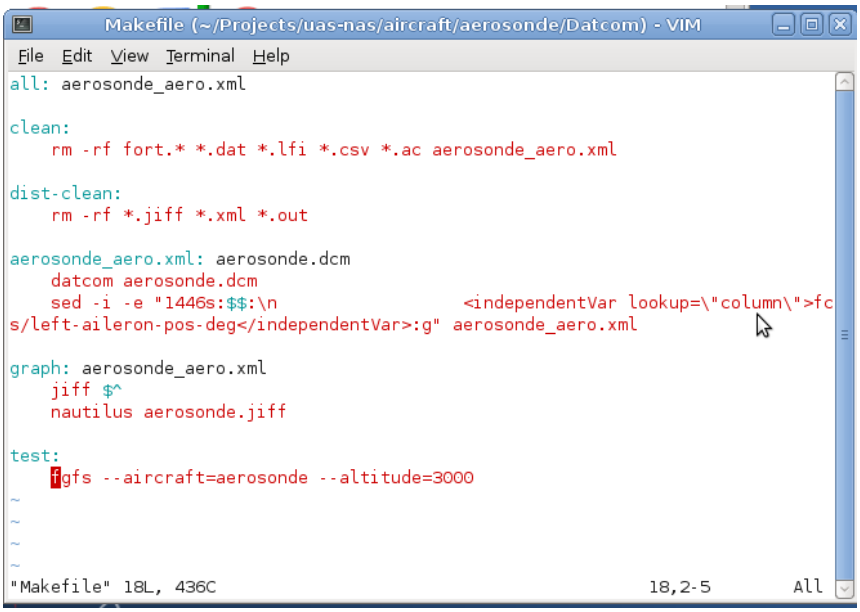

Figure 8: Content of Makefile in Datcom folder (vi Makefile)

The Makefile file in Datcom folder helps you to create various commands for the program. For example, typing “make graph” in the terminal will create the aircraft_name_aero.xml file, create the plots, and open the folder of the plots. This knowledge will speed up the time during debugging and testing.

Notes:

1. The program is very sensitive, thus when editing it is advisable to compile/make after a slight changes.
2. Most of the input is self-explanatory and the basic descriptions for the inputs are given as comment.
3. Main variable that was confusing were between SSPNE and SSPN. The difference is shown in the figure below where b*/2 is SSPNE and SSPN is b/2.

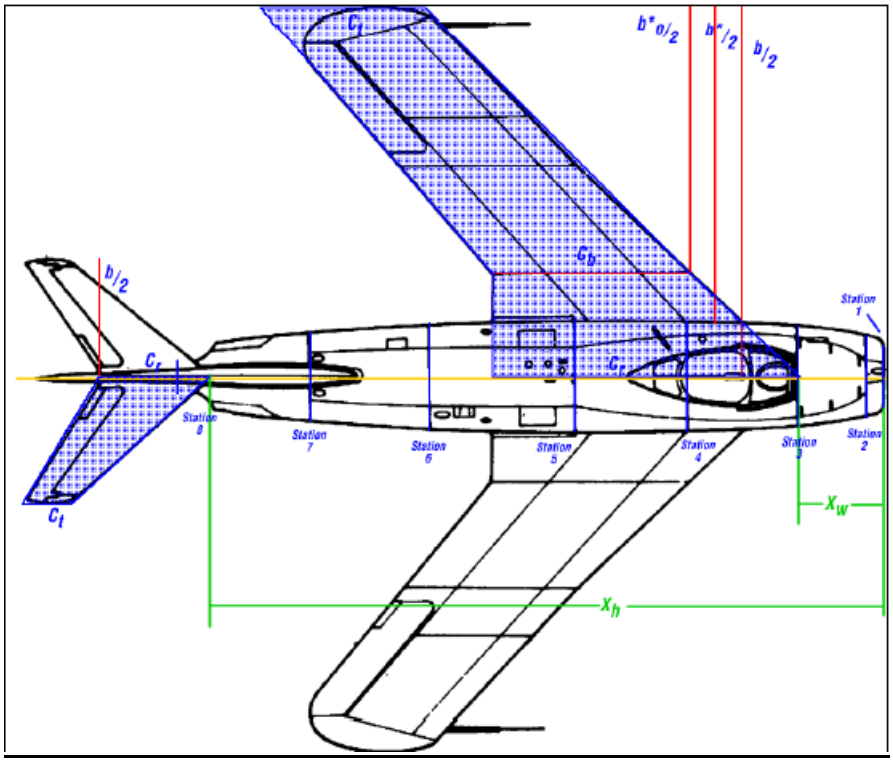

Figure 8: Aircraft with dimensions (SSPNE & SSPN)

4. DATCOM+ does not produce all output without all the basic part of the airplane above. Hence for a V-tail aircraft (for example shadow and aerosonde), a vertical tail were added into the.

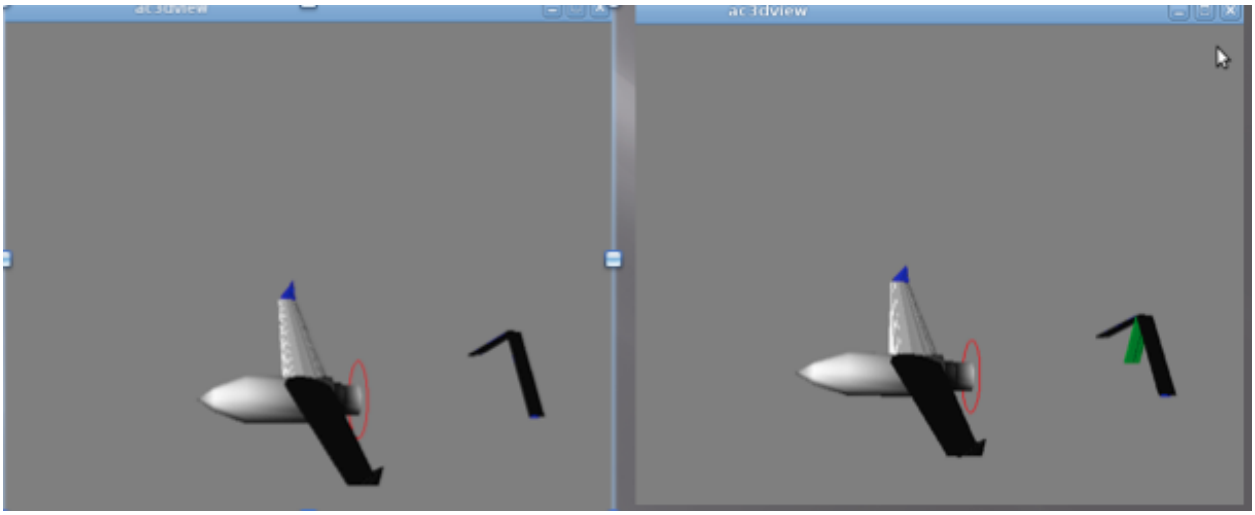

Figure 9: Aerosonde with (on the right) and without (on the left) vertical tail.

As can be seen from figure 9, vertical tail is added on the right. Although it does not present the best representation of Aerosonde, this addition of vertical tail is necessary because Datcom doesn't accept dihedral angle horizontal tail or V-tail in the computation of some of the aerodynamics properties. With the addition of V-tail, the replicated aerodynamics graphs are more logical.

5. To have a guideline on what graphs output should look like, compare the graphs shape and values with the example given by DATCOM+ which is shown as below (see figure 10). To confirm the outputs of the aerodynamics graphs, you are suggested to make a comparison between the aerodynamics graphs of the UAV that is being worked on and the successfully constructed UAVs.

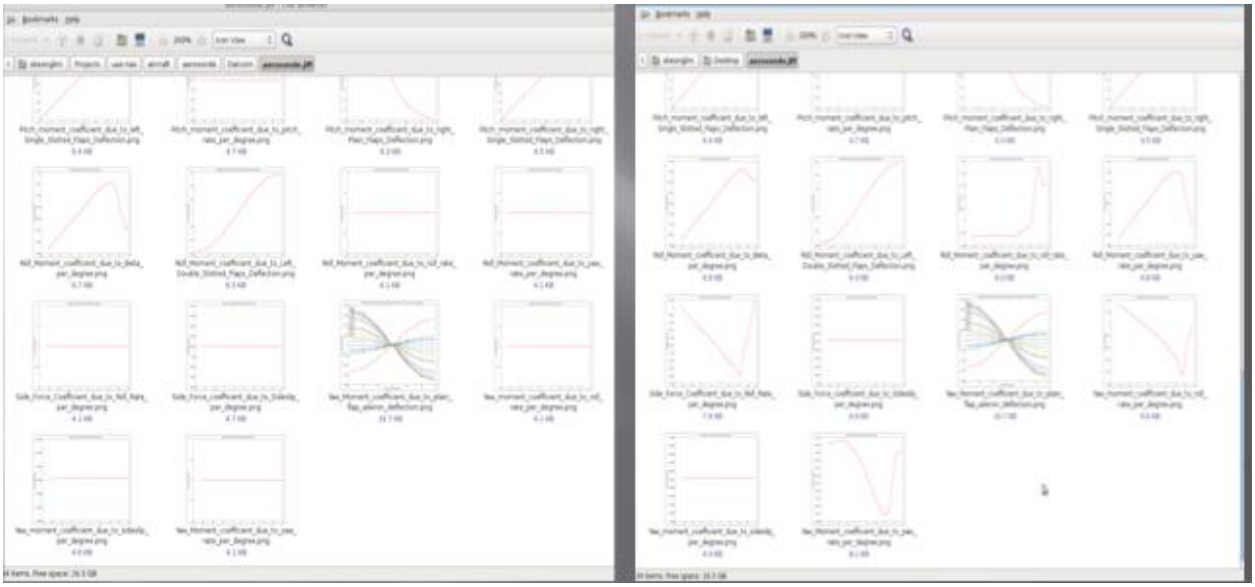

Figure 10: Sample aerodynamics graph (on the right). Aerodynamics graphs without V-tail (on the left).

From figure 10, the removal of the V-tail results in getting a lot of not remarkable outputs. The focus here is on the roll-moment coefficient and side force coefficient plot. The differences between the other plots are not considered because the correlations of the mismatching of the other plots remain in doubt.

<u>Difference between the model output in Blender and FlightGear</u>

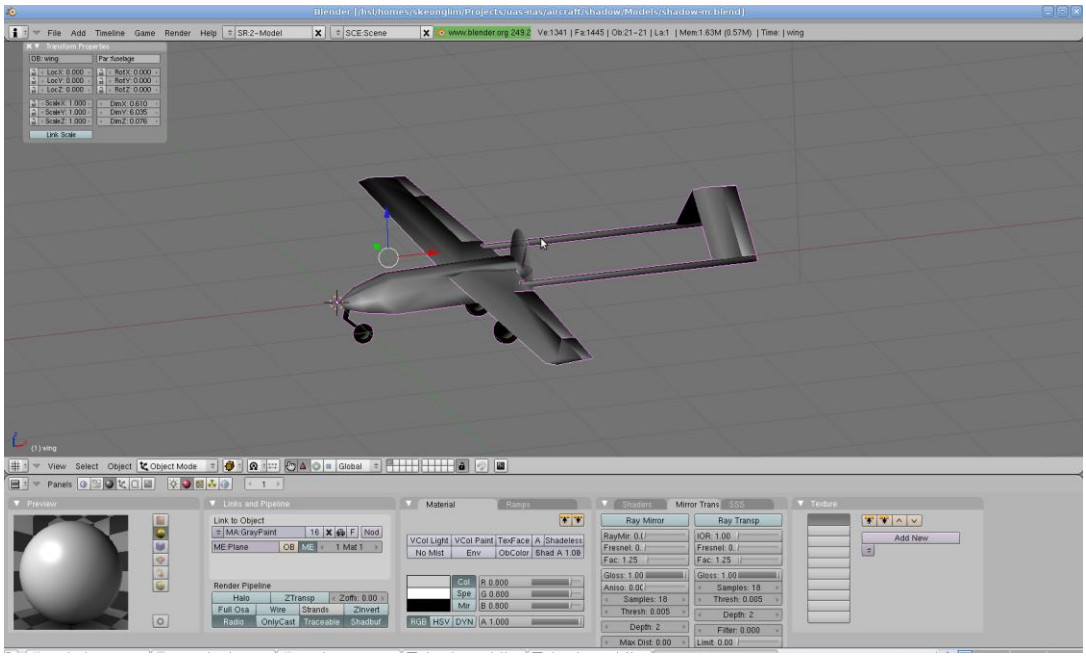

Figure 11: Aircraft model in Blender (Shadow)

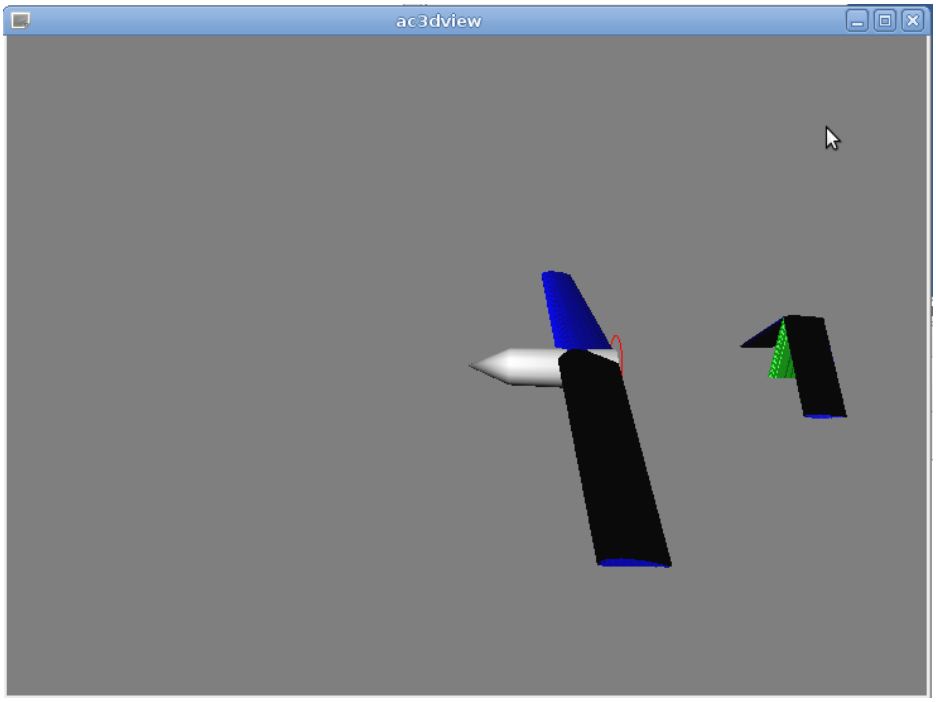

Figure 12: Aircraft model in Datcom (Shadow)

From the figures above (figure 11 and 12), there are differences between the model that Datcom produces based the data inputted into the program and the model which was made in Blender. The Datcom's model does not have boom and wheels, and the fuselage is not an exact representative of the real model. This is due to limited data that could be entered into Datcom.

Nonetheless, the model that produced by Datcom should be a close representative on defining the aerodynamics of the aircraft (shadow). The wings' dimensions, tails' dimensions, and the size of the fuselage generally govern the aerodynamic flow of air of the aircraft should be accurate to the real model.

Engines

The engine information that is inputted to the aircraft_name.xml is obtained from Aeromatic website http://jsbsim.sourceforge.net/aeromatic2.html . This website is used because it produces the engine configurations files that are usable by the JSBSim flight dynamic model.

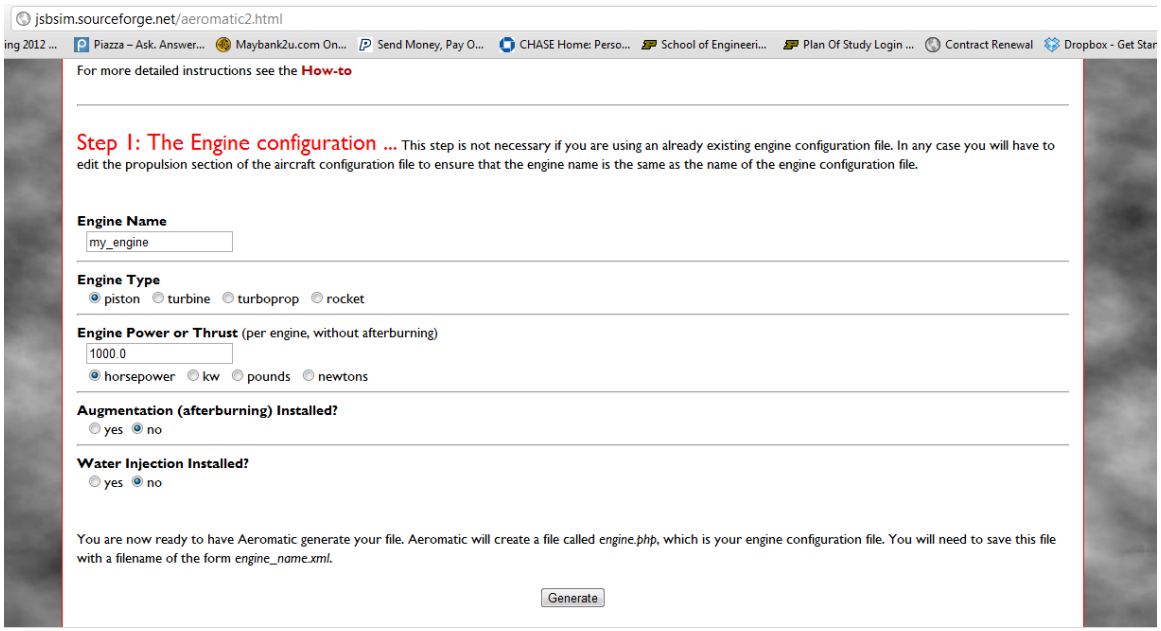
jsbsim.sourceforge.net/aeromatic2.html
For more detailed instructions see the How-to
Step 1: The Engine configuration ... This step is not necessary if you are using an already existing engine configuration file. In any case you will have to edit the propulsion section of the aircraft configuration file to ensure that the engine name is the same as the name of the engine configuration file.
Engine Name
my_engine
Engine Type
piston turbine turboprop rocket
Engine Power or Thrust (per engine, without afterburning)
1000.0
horsepower kw pounds newtons
Augmentation (afterburning) Installed?
yes no
Water Injection Installed?
yes no
You are now ready to have Aeromatic generate your file. Aeromatic will create a file called engine.php, which is your engine configuration file. You will need to save this file with a filename of the form engine_name.xml.
Generate

Figure 13: "Aeromatic" website

Engine configuration and Propeller configuration are required for the aircrafts that have been modeled in this semester. Propeller configuration is needed only if the aircraft has propeller. Some of the information that is required for the generation of the engine configurations are engine type, engine power or thrust, augmentation installed (yes or no), and water injection installed (yes or no). On the other hand, the engine power, maximum engine rotation speed (RPM), pitch (fixed or variable), and the propeller diameter* are required in order to generate the propeller configuration.

*Remark: Propeller diameter has to be in "feet". The other options such as "meter" or "inch" will produce inaccurate outcomes.

FlightGear

FlightGear is a free open source flight simulation that can be downloaded at http://www.flightgear.org/download/. FlighGear is used for the flight path simulation of the desired aircrafts in this semester. The steps of adding the desired aircraft to FlightGear, and the checks of flight properties in FlightGear simulator are discussed by parts as below.

1. Linking Process.
2. Modification of the .fgfsrsc file.
3. Start of FlighGear.
4. Checks of aircraft in FlightGear.

Linking Process

All sketching, modeling, and changing of the aircrafts' properties are done in a different folder than FlightGear. A symbolic link shall be made between the aircraft folder in FlightGear and our working folder in order to link them together. With such linkage, changes made on the aircrafts are reflected on the models that will be used in FlightGear simulator directly.

The linking process can be done by typing the following commands in Terminal. * Commands to do symbolic link.

Modification of .fgfsrc file

Aircrafts that will be simulated in FlighGear are stored at .fgfsrc file. Modification on that file shall be made in order to get the correct aircraft in place. It can be done by typing the following command in the Terminal:

1) vi .fgfsrc
2) '#' the undesirable aircraft, add the name of the desirable aircraft on the file.
3) Save the file.

Details:

1. "vi .fgfsrc" to open the .fgfsrc file to change the aircraft to be represented in FlightGear.
2. Commented out the undesired aircraft by inserting "#" before the aircraft. Typed in the new aircraft's name in the list.
3. Saves and closes the .fgfsrsc file

Start of FlightGear

The selected aircraft will be ready to be visualized in FlightGear by typing "fgfs" in the Terminal. Four interested aspects of the aircraft are attempted to observe from the FlightGear simulator this semester. These include animations, view properties, takeoff conditions, and flying conditions are focused in this semester.

FlighGear may be strange for most of the undergraduate students because of its high frequent use of keyboard keys. It incorporates many shortcut keys to perform some of the functions. In order to allow future groups can be more easily learn the vital function keys in FlightGear. A list of shortcut keys is provided below as references. A full list of shortcut keys can be also retrieved at http://wiki.flightgear.org/Keyboard_shortcuts

Table 2: Useful Shortcut Keys in FlightGear

V	Change of views
P	Pause
X	Zoom in
Ctrl+x	Return to default view
H	Head Up Display
]	Flaps down
[	Flaps up
Shift+] →s	Ignite the engine

Animations

Graphical movements on the desirable components shall be checked before any other actions are taken. If some of the desirable movements aren't performed as expected, users shall first check the names and values inputted in aerosonde-model.xml to seek for the differences in the names and values than aerosonde.blend. If such approach is not successful, users shall check for the offsets in the dimensions that are shown in aerosonde.blend and the provided spreadsheet.

View Properties

View properties shall be checked to make sure the controls of each control surface are in well conditions. It is important for users to get an access to view the internal properties of the aircraft such as the controls, aerodynamics, orientations, and the translations of the aircraft (see figure14).

Figure 14: View Properties in FlightGear

Takeoff Condition

One of the key features that we are observing on FlightGear is the takeoff condition of the aircraft. After making sure the aircraft properties are correct, we have to test its take off ability and observe the behavior and flight pattern when the aircraft is off from the ground to the air. Aircraft shall be takeoff nicely with the correct Aerodynamics and engine applied. If a smooth take off condition and flying pattern was not met, we are required to check the input values in the Datcom and adjust the values accordingly to get a good takeoff condition.

Dropping Condition

Well flying Aerosonde is expected to fly in a smooth pattern. Smooth pattern is defined as the periodic curve liked movement. Aerosonde will dive down to seek for acceleration and be raised up by gaining enough velocity and lift. The aerodynamics of the aircraft shall be first checked if such flying pattern is not met. Input values in aircraft.dcm shall also be checked and adjusted to meet such flying requirements. However, there is always a tradeoff between the smooth flying pattern and the trimmed conditions of the aircraft.

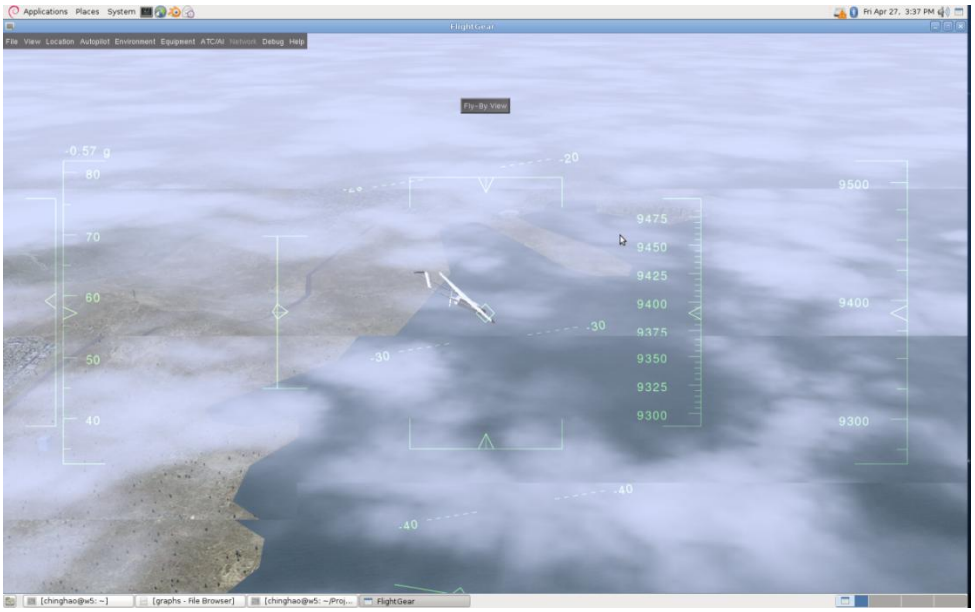

Figure 15: Aircraft in smooth descent (Aerosonde).

We shall better understand the trimmed conditions of the aircraft so that we get the best balance between the trimmed conditions and the smooth flying pattern.

Remark: In order to get a smooth flight, we change the center of gravity accordingly based on the rule of thumb. The change in the cg of the aircraft is advisable for future group in order to get a good balance between the trimmed speed and the smooth flying pattern.

JSBSIm

Introduction

JSBSim is a library that can be called, supplied with inputs (such as control inputs from the pilot), and returning outputs (describing the aircraft's state at any moment in time). This software would work for both windows and linux. However, this software works best with l linux system (Debian) as it has a terminal. This is as when trying to find the output, the iteration can be observed in the terminal thus would know the problem easier.

Procedure

1) Insert path location of engine, system, and model
2) Aircraft condition (cruise initial speed..)
3) Guess trim condition
4) Converge values
5) The converge values can be made
6) Obtaining data (the figure below shows a sample output data of an aircraft (shadow):
    a. Ensure weight and flight condition
    b. Cruise data (normally empty weight, mid weight and full weight)
    c. Maximum throttle when climbing (to find flight path angle)
    d. Min throttle when descent (to find flight path angle)

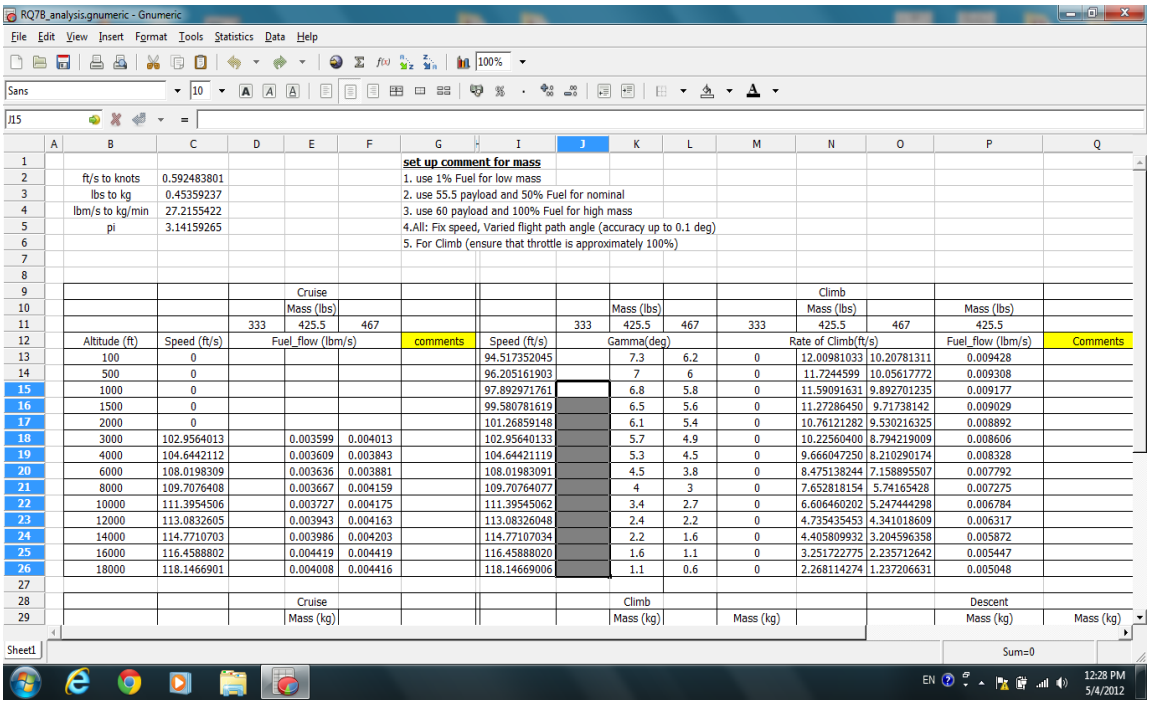

Figure 16: Data Output (Shadow)

Insert path location of engine, system, and model

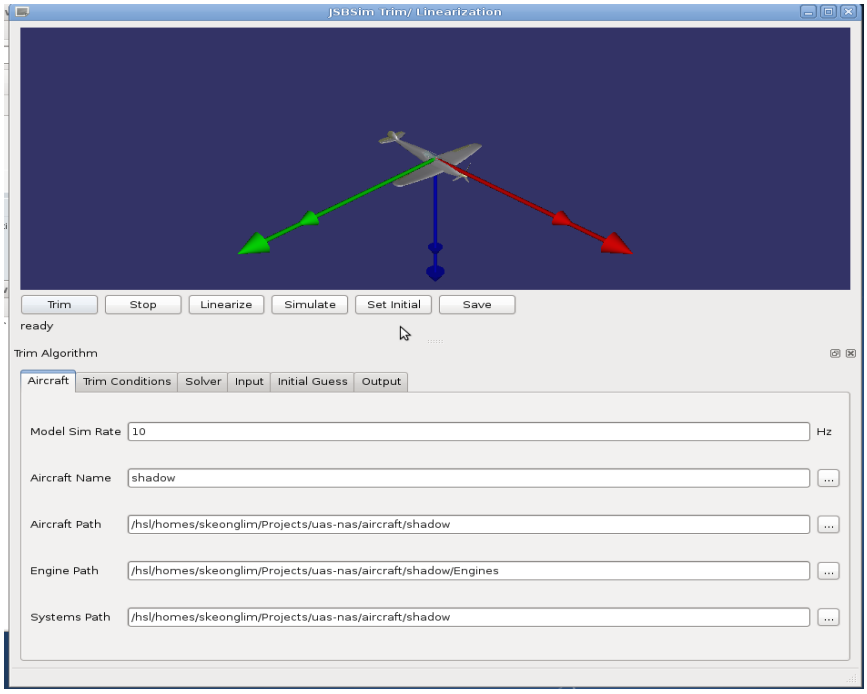

Figure 17: JSBSim in Aircraft Path view

Figure 17 shows the location of the aircraft path, engine path and system path which needs to be set.

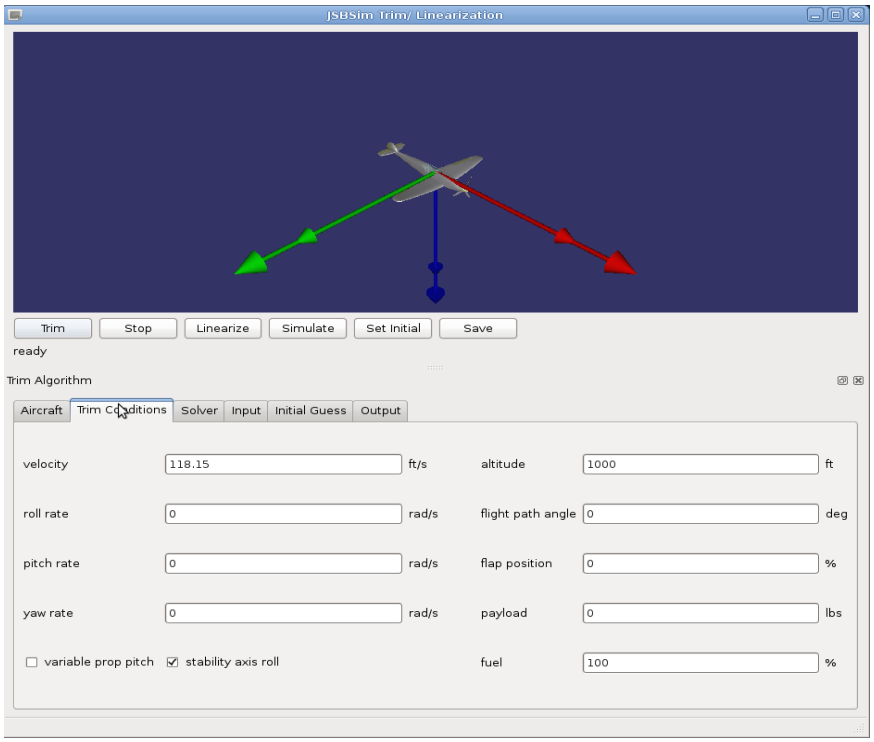

Figure 18: JSBSim in Trim Condition view

Figure 18 depicts where the inputs are put for desired flight condition (cruise, climbing and descent). The parameters that defines the flight condition includes velocity, altitude, flight path angle, flap position, payload and percentage fuel. The example shows an empty payload aircraft cruising at 1000ft with 110.15 ft/s.

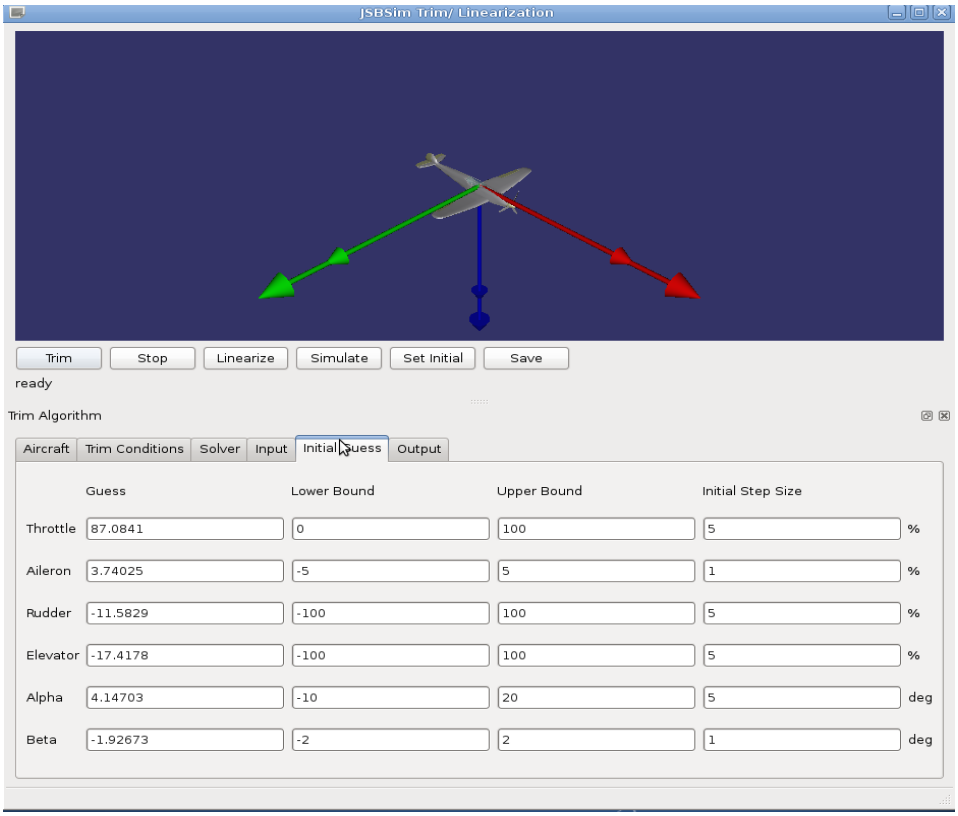

Figure 19: JSBSim in Initial Guess view

Figure 19 shows the initial guess values that the user can input. The closer the input to the output conditions, the faster JSBSim will trim. The lower and upper bound would control the limit of the 6 Degree of Freedom based on the aircraft specification. A good practice that is recommended for a faster trimming time is to set the initial after each run of the trim. This facilitates the trimming time because the initial guess values that are reset are much closer to the exact trim values than before.

Once, the trim button at the top is clicked, the program will run and the terminal will show as follow:

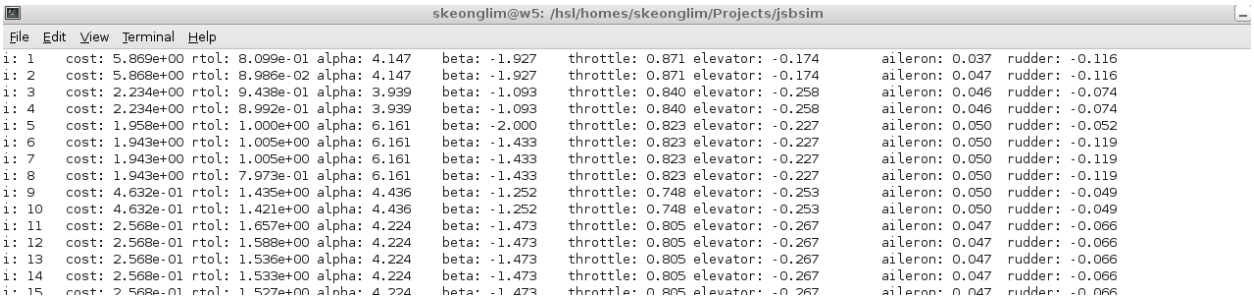
skeonglim@w5: /hsl/homes/skeonglim/Projects/jsbsim
File Edit View Terminal Help
i: 1 cost: 5.869e+00 rtol: 8.099e-01 alpha: 4.147 beta: -1.927 throttle: 0.871 elevator: -0.174 aileron: 0.037 rudder: -0.116
i: 2 cost: 5.868e+00 rtol: 8.986e-02 alpha: 4.147 beta: -1.927 throttle: 0.871 elevator: -0.174 aileron: 0.047 rudder: -0.116
i: 3 cost: 2.234e+00 rtol: 9.438e-01 alpha: 3.939 beta: -1.093 throttle: 0.840 elevator: -0.258 aileron: 0.046 rudder: -0.074
i: 4 cost: 2.234e+00 rtol: 8.992e-01 alpha: 3.939 beta: -1.093 throttle: 0.840 elevator: -0.258 aileron: 0.046 rudder: -0.074
i: 5 cost: 1.958e+00 rtol: 1.000e+00 alpha: 6.161 beta: -2.000 throttle: 0.823 elevator: -0.227 aileron: 0.050 rudder: -0.052
i: 6 cost: 1.943e+00 rtol: 1.005e+00 alpha: 6.161 beta: -1.433 throttle: 0.823 elevator: -0.227 aileron: 0.050 rudder: -0.119
i: 7 cost: 1.943e+00 rtol: 1.005e+00 alpha: 6.161 beta: -1.433 throttle: 0.823 elevator: -0.227 aileron: 0.050 rudder: -0.119
i: 8 cost: 1.943e+00 rtol: 7.973e-01 alpha: 6.161 beta: -1.433 throttle: 0.823 elevator: -0.227 aileron: 0.050 rudder: -0.119
i: 9 cost: 4.632e-01 rtol: 1.435e+00 alpha: 4.436 beta: -1.252 throttle: 0.748 elevator: -0.253 aileron: 0.050 rudder: -0.049
i: 10 cost: 4.632e-01 rtol: 1.421e+00 alpha: 4.436 beta: -1.252 throttle: 0.748 elevator: -0.253 aileron: 0.050 rudder: -0.049
i: 11 cost: 2.568e-01 rtol: 1.657e+00 alpha: 4.224 beta: -1.473 throttle: 0.805 elevator: -0.267 aileron: 0.047 rudder: -0.066
i: 12 cost: 2.568e-01 rtol: 1.588e+00 alpha: 4.224 beta: -1.473 throttle: 0.805 elevator: -0.267 aileron: 0.047 rudder: -0.066
i: 13 cost: 2.568e-01 rtol: 1.536e+00 alpha: 4.224 beta: -1.473 throttle: 0.805 elevator: -0.267 aileron: 0.047 rudder: -0.066
i: 14 cost: 2.568e-01 rtol: 1.533e+00 alpha: 4.224 beta: -1.473 throttle: 0.805 elevator: -0.267 aileron: 0.047 rudder: -0.066

Figure 20: Trimming (running in Terminal)

When, it is able converged the result will be shown and all the output values with initial conditions are shown in the terminal. In general, the cost will converge when its value is at the power of negative 2 (10^{-2}). The figure below depicts the output.

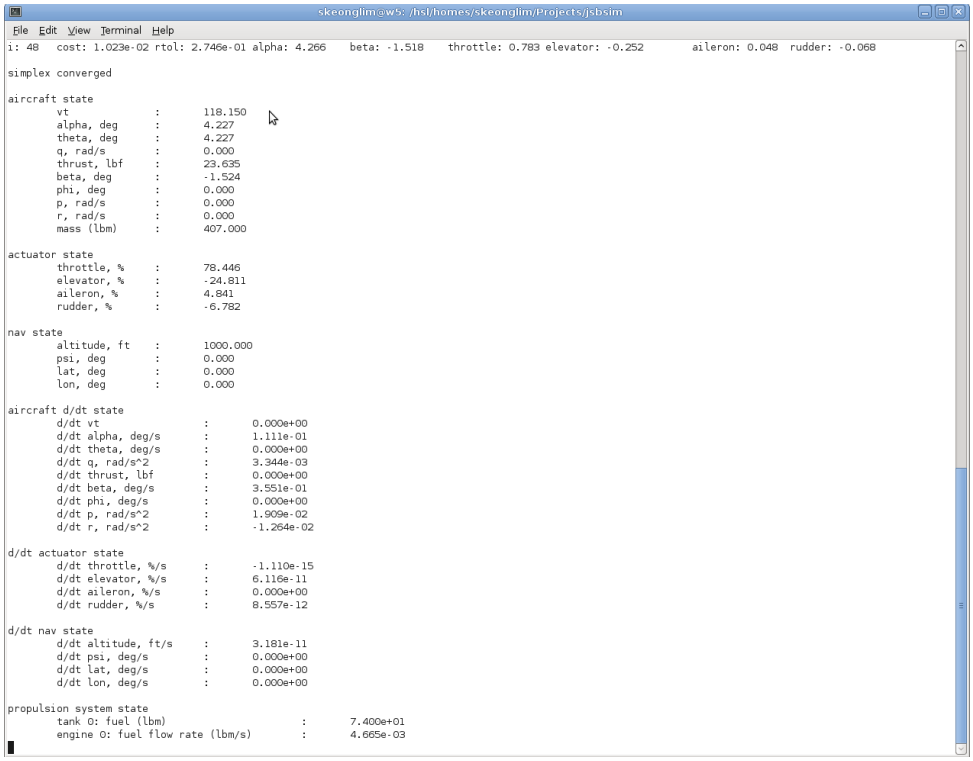
skeonglim@w5: /hsl/homes/skeonglim/Projects/jsbsim
File Edit View Terminal Help
i: 48 cost: 1.023e-02 rtol: 2.746e-01 alpha: 4.266 beta: -1.518 throttle: 0.783 elevator: -0.252 aileron: 0.048 rudder: -0.068
simplex converged
aircraft state
vt : 118.150
alpha, deg : 4.227
theta, deg : 4.227
q, rad/s : 0.000
thrust, lbf : 23.635
beta, deg : -1.524
phi, deg : 0.000
p, rad/s : 0.000
r, rad/s : 0.000
mass (lbm) : 407.000
actuator state
throttle, % : 78.446
elevator, % : -24.811
aileron, % : 4.841
rudder, % : -6.782
nav state
altitude, ft : 1000.000
psi, deg : 0.000
lat, deg : 0.000
lon, deg : 0.000
aircraft d/dt state
d/dt vt : 0.000e+00
d/dt alpha, deg/s : 1.111e-01
d/dt theta, deg/s : 0.000e+00
d/dt q, rad/s^2 : 3.344e-03
d/dt thrust, lbf : 0.000e+00
d/dt beta, deg/s : 3.551e-01
d/dt phi, deg/s : 0.000e+00
d/dt p, rad/s^2 : 1.909e-02
d/dt r, rad/s^2 : -1.264e-02
d/dt actuator state
d/dt throttle, %/s : -1.110e-15
d/dt elevator, %/s : 6.116e-11
d/dt aileron, %/s : 0.000e+00
d/dt rudder, %/s : 8.557e-12
d/dt nav state
d/dt altitude, ft/s : 3.181e-11
d/dt psi, deg/s : 0.000e+00
d/dt lat, deg/s : 0.000e+00
d/dt lon, deg/s : 0.000e+00
propulsion system state
tank 0: fuel (lbm) : 7.400e+01
engine 0: fuel flow rate (lbm/s) : 4.665e-03

Figure 21: JSBSim output

Remark:

1) A descent model of the aircraft would want the elevator condition/output to be as close to zero during cruising. Hence, the best guideline to judge the model would be obtaining a low elevator deflection when cruising.

Problem may be encountered and the recommended solutions

(Note: "*" : Use this way of solving only if you had double checked that all inputted values are correct and accurate to the data given, and the DATCOM output has been generated correctly)

2) Obtaining all the graphs output of DATCOM correctly (compared with example):
    a. Checked all the input values again
    b. Ensure all the dimensions are show in the AC3D view in DATCOM+
    c. Make sure all the basic part of aircraft are present (example is V-tail)
    d. IF fail to find the error, redo by copying small part of the aircraft and remake the model part by part to identify the problem.
3) Aircraft does not fly right in FlightGear:
    a. Change cg value
    b. Change the Moment of Inertia in aircraft_name.xml. *
4) JSBSim does not converge:
    a. Check which values are max out; if aileron are max, increase surface area in aileron in DATCOM before Blender
    b. Change the CG till it converge but ensure that the model still fly right in FlightGear
5) JSBSim is very slow:
    a. Use a lower Ixx value for engine propeller *
    b. Change the Hz requirements of the program *

Review of Aircraft Models (Spring 2012)

i) Missile (SM-3)

Figure 22: Missile (SM-3)

Missile is modified based on the X15 model, the difference between this model than the other aircraft models is that Missile uses Missile Datcom to generate the aerodynamics properties instead of Datcom + that is used by other aircraft models.

ii) Orbiter

Figure 23: Orbiter

The orbiter model in Blender is drawn from scratch. The main difference between the outlines of orbiter than the other models is the control surfaces. Orbiter only has ailerons at the trailing edges of its wing. Orbiter has to take the roll and pitch control in one surface control because Orbiter doesn't have a tail.

iii) Aerosonde

Figure 24: Aerosonde

The 3D model of Aerosonde in Blender was built on top of Shadow. There were many problems during modeling. The methods to handle the errors were determined based on trials and errors were:

1) Flight is unstable when dropped in FlighGear. After trying to search for errors, w found out that the aerodynamics graphs are not the same as the standard aerodynamics graphs.
2) The flight pattern of Aerosonde in FlightGear was sill very unstable. We changed the roll moment of inertia of Aerosonde in order to increase Aerosonde's stability in x-direction.
3) We later observed that Aerosonde didn't have enough maneuverability and controls. We increased the sizes of the controls surfaces such as ailerons, flaps, rudders in order to increase our control of Aerosonde in FlightGear.
4) After achieving well flight path in FlightGear, we relized that Aerosonde couldn't be trimmed in JSBSim. We later figured out the reason was due to the insufficient output thrust of the engine that was estimated by Aeromatic. We then increased the horsepower of the engine in Aeromatic in order to reproduce a engine that matches the output thrust given by the company.
5) We were able to get Aerosonde to be trimmed in JSBSim. However, the trimmed conditions show a high elevator deflection during cruise. We change the position of the center of gravity (cg) of Aerosonde in order to solve this issue.

iv) Shadow

Figure 25: Shadow

Shadow was not able to trim to the given ceiling height that was provided as a requirement by the company. Also, Shadow has a much slower take-off speed than the information given by the company. A vertical is added on the modeling of Shadow in Datcom in order to generate the logical outcome of the roll moment coefficient plot

Flight Performance

Easystar – This model is not a model that was accomplished in this semester but should be used for comparison as it has the best flight path among all of the other aircraft models.

Orbiter – Good flight path when dropped from the certain altitude above the ground and has a good maneuverability.

Aerosonde – Unstable flight path in yaw and pitch direction. The instability in the yaw direction is due to the increase of the moment of inertia in x direction (Ixx), and the instability in the pitch direction is due to the change in the location of the center of gravity. However, it still represents a good maneuverability

Shadow – Unstable in pitch direction due to the change in the location of the center of gravity. The maneuverability is considered as controllable.

Future Improvements

Changes that were done to previous models which are inaccurate and need to be rechecked:

a) Shadow
    a. Increase engine stroke to reach ceiling height
b) Aerosonde
    a. Increased Ixx value of aircraft's model
    b. Increased horsepower of aircraft (Supposed to be 6hp but 20hp were used)

Main Issue that need to be checked:

a) Engines generated by Aeromatic website are inaccurate

Future Model:

a) Cargo UAS:
    a. Copied and changed name and path name from Shadow
    b. Blender model has been done by James Goppert

Conclusion

The discussions act as a reference for the future group. Instructions mentioned are based on the rule of thumb. If such results replicated above can not be reproduce, users shall consult their instructors for a more precise analysis of the problems that they have encountered.